Optimal Distributed Multicast Routing using Network Coding: Theory and Applications

Yi Cui, Yuan Xue, Klara Nahrstedt
Department of Computer Science, University of Illinois at Urbana-Champaign
{*yicui, xue, klara*}@cs.uiuc.edu

***Abstract*— Multicast is an important communication paradigm, also a problem well known for its difficulty (NP-completeness) to achieve certain optimization goals, such as minimum network delay. Recent advances in network coding[1], [2] has shed a new light onto this problem. In network coding, forwarding nodes can perform arbitrary operations on data received, other than forwarding or replicating, to enhance throughput of a multicast session. In this paper, we show that with the aid of network coding, the once intractable optimal multicast routing problem becomes tractable. In this problem, given a set of multicast sessions and their traffic demands, one tries to route the multicast traffic regarding various objectives, such as to minimize overall delay, or to maximize the battery life of each node. We further show that this problem can be solved in a distributed fashion: each node makes its own routing decisions based on periodic updating information from neighboring nodes. We prove that starting from any initial routing assignment, the proposed distributed routing algorithm is able to converge to the point where the value of the objective function is optimized. Our solution can be fit into a variety of networks to achieve different optimization goals. The examples in this paper include minimum delay routing in overlay multicast, and maximum lifetime routing in multi-hop wireless network.**

I. Introduction

Optimal data routing in a network can be often understood as a multicommodity flow problem. Given a network and a set of commodities, i.e., a set of source-destination pairs, one tries to achieve certain optimization goal, such as minimum delay, maximum throughput, while maintaining certain fairness among all commodities. The constraints of such optimization problems are usually network link capacity and traffic demand of each commodity. Multicommodity flow problem has been well studied as a typical linear programming problem. Its distributed solutions have also been proposed[3][4].

However, when each commodity becomes a multicast session consisting of a source and several destination nodes, the same problem becomes intractable even in a centralized fashion. If the goal is to minimize network delay, it becomes the Steiner tree problem, which is NP-hard[5]. If the goal is to maximize achievable throughput, its difficulty is equivalent to packing Steiner trees, a problem even harder[6], [7], [8].

Recent advances on network coding has shed a new light onto this problem. Network coding generalizes traditional routing paradigm in which relaying nodes can only forward or replicate, by allowing them to perform arbitrary operation on information received to generate output. It is proved [1][2] that with network coding, the achievable throughput of a multicast session is the minimum of the maximum flow from the sender to any receiver.

Given network coding's amazing ability at improving throughput of existing network, we do not consider it the most wanted feature, since throughput is not the most urgent issue at the current moment. With the rapid advancement of optical fiber and ultraband technologies, the present (and ever growing) capacities of both wireline and wireless networks can easily satisfy the demands of current applications.

What we consider the biggest advantage of network coding is the discovery that it makes the once intractable *optimal multicast routing* problem tractable. Furthermore, we show that this problem can be solved in an entirely distributed fashion. The problem is roughly defined as follows. Given a network, a set of multicast sessions, each with their own traffic demands, we try to route the multicast traffic regarding various objectives, such as to avoid congestion, minimize overall delay, or to maximize the battery life of each node in a wireless network. The rigorous definitions of these objective functions are given in Sec. III.

With the aid of network coding, we are able to formulate the optimal multicast routing problem in the fashion of multicommodity flow (details in Sec. II). The major contribution of this work is an optimal distributed solution to the same problem. In this solution, each node makes its own routing decisions based on periodic updating information from neighboring nodes. More importantly, starting from any initial routing assignment,

it should finally converge to the optimal point, such as minimum network delay. Our solution inherits the same design philosophy of Gallager's algorithm[3], but is significantly different from it, since we try to achieve optimal routing in the setting of multicast communication with network coding.

Although our solution is general enough to fit into a variety of networks to achieve different optimization goals, we consider it more suitable to be employed in a new generation of application-level networks, such as overlay network, multi-hop wireless network, etc. In these networks, each node is flexible enough to be configured to perform various operations. In contrast, traditional Internet is already too rigid to even turn on its multicast switch, not to mention adding new functionalities such as network coding.

Moreover, multicast plays an essential role in the most popular applications supported by these new networks, such as overlay multicast[9], P2P content distribution, wireless sensor network, etc. They often has a system goal such as lifetime maximization in sensor network, i.e., to maximize the duration that all sensor nodes are up until one of them has its battery drained. Such a goal of "system optimization" is radically different to "user optimization", which is the goal of most current networking algorithms. In Internet, each node attempts to send each packet over a route that minimizes that packet's delay with no regard to other packet's delays. If the same analogy is applied to sensor network routing, each node would attempt to minimize the amount of energy it spends on each packet transmitted, which deviates from the system optimization goal of maximum lifetime.

We believe our solution is well suited to achieve the goal of "system optimization" in the above mentioned network and application settings. Due to space constraint, we only report theoretical result of our work.

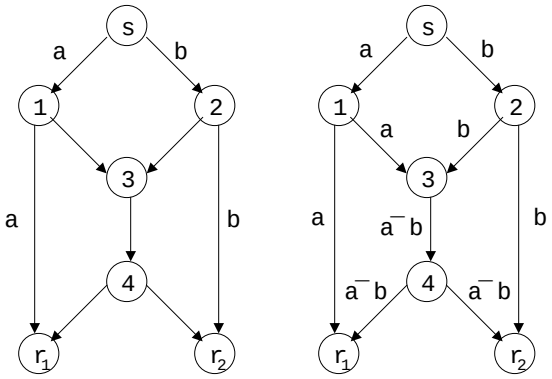
The rest of this paper is organized as follows. We briefly go over the concept of network coding and present our network model in Sec. II. In Sec. III, we first discuss necessary and sufficient conditions to achieve optimal routing in the general network model, then illustrate two particular examples: minimum-delay routing in overlay network and maximum-lifetime routing in multi-hop wireless network. Sec. IV presents our distributed routing algorithm and proves that it is able to converge to the point where the value of the objective function is optimized. Sec. V discusses some practical issues. Finally, Sec. VI concludes the paper.

II. Preliminaries

A. Network Coding: The Concept

An example to illustrate the concept of network coding is shown in Fig. 1. Consider a directed network in which each link has identical capacity. We have a multicast session where S is the sender, R_1 and R_2 are receivers. a and b represent two independent information flows originating from the sender S. As shown in Fig. 1 (b), node 3 transmits the coded flow $a \oplus b$ along the "bottleneck" link $(3, 4)$ to node 4, which in turn forwards the coded flow to receivers r_1 and r_2, which can recover $\{a, b\}$ from $\{a, a \oplus b\}$ and $\{b, a \oplus b\}$. On the other hand, without network coding (Fig. 1 (a)), receivers r_1 and r_2 can only receive one of the two flows.

It is proved by Ahlswede et al.[1] that with network coding, the achievable throughput of a multicast session can be acquired by running max-flow algorithm from the source to each individual receiver, then choosing the minimal result. Koetter et al.[2] prove the same result using algebraic approach. Li et al.[10] further shows that the above result can be obtained by running linear coding. Chou et al.[11] are the first to propose a practical network coding solution.



(a) Throughput without Network Coding

(b) Throughput with Network Coding

Fig. 1. The Effects of Network Coding

B. Network Model

We consider a n-node network, where the nodes are represented as $\mathcal{N} = \{1, 2, \ldots, n\}$. Let $\mathcal{L}$ be the set of links, denoted as $\mathcal{L} = \{(i, k) \mid \text{a link goes from } i \text{ to } k\}$. Each link (i, k) is associated with a capacity C_{ik}. There are a set of multicast sessions $\mathcal{M}$. For each session $m \in \mathcal{M}$, it has a sender $S(m)$, and a set of receivers $R(m)$.

Let $r_i^m(j) \geq 0$ be the traffic of session m, in bits/s, generating at node i and destined for node j (data sink). $r_i^m(j) > 0$ only if node $i = S(m)$, and $j \in R(m)$, i.e., if node i is the sender of session m, and node j is one of

the receivers of m. We also define node flow $t_i^m(j)$ to be the total traffic of session m at node i destined for node j. $t_i^m(j)$ includes both $r_i^m(j)$ and the traffic from other nodes that is routed through i to destination j. Finally, $\phi_{ik}^m(j)$ is the fraction of the node flow $t_i^m(j)$ routed over link (i,k). It is always true that $\phi_{ik}^m(j) = 0$ if $(i,k) \notin \mathcal{L}$ (no traffic can be routed through non-existent link), or $i = j$ (traffic that has reached its destination is not sent back into the network). Also, node i must route its entire node flow $t_i^m(j)$ through all links, i.e.,

$$\sum_{k \in \mathcal{N}} \phi_{ik}^m(j) = 1, \forall i, j \in \mathcal{N}, \forall m \in \mathcal{M} \tag{1}$$

Now we express the relation of above notations as follows:

$$t_i^m(j) = r_i^m(j) + \sum_{l \in \mathcal{N}} t_l^m(j)\phi_{li}^m(j), \forall i, j \in \mathcal{N}, \forall m \in \mathcal{M} \tag{2}$$

Eq. (2) expresses flow conservation: for a given multicast session s, the traffic into a node for a given destination is equal to the traffic out of it for the same destination.

Lemma 1 Given the input set $\boldsymbol{r}$ and routing variable set $\boldsymbol{\phi}$, the set of equations (2) has a unique solution for $\boldsymbol{t}$. Each element $t_i(j)$ is nonnegative and continuously differentiable as a function of $\boldsymbol{r}$ and $\boldsymbol{\phi}$.

For each session s, we define the amount of traffic on link (i,k) as the union of all flows through it.

$$f_{ik}^m = \max_j t_i^m(j)\phi_{ik}^m(j), \forall (i,k) \in \mathcal{L} \tag{3}$$

According to Alhswede et al.[1], for a given input set $\boldsymbol{r} = \{r_i^m(j) \mid i, j \in \mathcal{N}, m \in \mathcal{M}\}$, if there exists a routing solution $\boldsymbol{\phi} = \{\phi_{ik}^m(j) \mid i, j, k \in \mathcal{N}, m \in \mathcal{M}\}$ that is feasible, i.e.,

$$f_{ik} = \sum_{m \in \mathcal{M}} f_{ik}^m \leq C_{ik}, \forall (i,k) \in \mathcal{L} \tag{4}$$

then the achievable throughput by network coding in each multicast session m is $\min_{j \in R(m)} r_{S(m)}^m(j)$. Furthermore, any feasible solution is schedulable by a network coding assignment.

Now we can formalize our "system optimization" goal according to the following format. For example, if the delay on each link (i,k) is a function of traffic on it, $D_{ik}(f_{ik})$, and our goal is to minimize the overall network delay, it can be formalized into the following optimization problem.

$$\begin{aligned} \textbf{D:} \quad & \textbf{minimize} \quad D = \sum_{(i,k) \in \mathcal{L}} D_{ik}(f_{ik}) \\ & \textbf{subject to} \quad (1), (2) (\textbf{flow constraint}) \\ & \qquad\qquad\quad (3) (\textbf{union of flow constraint}) \\ & \qquad\qquad\quad (4) (\textbf{capacity constraint}) \end{aligned}$$

III. Optimality Conditions for Distributed Multicast Routing

In this section, we analyze the optimality conditions for distributed multicast routing. We show that when the system delay D is minimized, within each session m, each node i, for a given receiver j, the partial derivative of D to the routing variable $\phi_{ik}^m(j)$ (marginal delay on link (i,k)) is the same for all links (i,k) originating from node i.

An analogy is that within an electrical network where each wire has different resistance, certain currents flow from the sender node to the receiver node. By Dirichlet principle, the potentials taken within the electrical network minimize the total energy dissipation. And when it happens, the potentials (partial derivative of energy dissipation to currents) of all wires sharing the same positive end are the same.

Sec. III-A goes through the formal analysis to reach the above result. Note that although we use delay as an example objective, the same conclusion holds for any type of objective function. In Sec. III-B and III-C, we show that with necessary adjustment to the network model and objective function, we can derive the same optimality conditions for multicast routing in a wide spectrum of problem settings. Our examples include minimum delay routing in overlay multicast, and maximum lifetime routing in multi-hop wireless network.

A. General Model

We calculate the partial derivatives of the delay D with respect to the inputs $\boldsymbol{r}$ and the routing variables $\boldsymbol{\phi}$. We first consider $\partial D / \partial r_i^m(j)$. Assume a small increment ϵ in the input $r_i^m(j)$. For each adjacent node k, an increment $\epsilon\phi_{ik}^m(j)$ of this new incoming traffic will flow over (i,k), and to first order, this will cause an increment delay on that link of

$$\epsilon\phi_{ik}^m(j) D'_{ik}(t_i^m(j)\phi_{ik}^m(j))$$

where

$$D'_{ik}(t_i^m(j)\phi_{ik}^m(j)) = \frac{dD_{ik}(f_{ik})}{df_{ik}} \cdot \frac{df_{ik}}{df_{ik}^m} \cdot \frac{df_{ik}^m}{d(t_i^m(j)\phi_{ik}^m(j))}$$

$D'_{ik}(t_i^m(j)\phi_{ik}^m(j))$ can be calculated as follows. A commonly used link delay function is defined by Kleinrock[12] as follows.

$$D_{ik}(f_{ik}) = \frac{f_{ik}}{(C_{ik} - f_{ik})} \tag{5}$$

This function assumes that queueing delays are the only noneligible source of delay in a network, and each link traffic can be modelled as Poisson message arrivals with independent exponentially distributed lengths. In fact, we do not need to know what $D_{ik}(f_{ik})$ is, as long as this function is increasing and convex in f_{ik}. In practice, we can also choose to directly measure D_{ik} and its derivative, which we will discuss in Sec. V-A.

According to Eq. (4), $df_{ik}/df_{ik}^m = 1$. According to Eq. (3),

$$\frac{df_{ik}^m}{d(t_i^m(j)\phi_{ik}^m(j))} = \begin{cases} 1/n & \text{if } t_i^m(j)\phi_{ik}^m(j) \text{ and } n-1 \\ & \text{other flows on link}(i,k) \\ & \text{are the maximum} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

If node k is not the destination node, then the increment $\epsilon\phi_{ik}^m(j)$ of extra traffic at node k will cause the same incremental delay onward as an increment $\epsilon\phi_{ik}^m(j)$ of new input traffic at node k. To first order this incremental delay will be $\epsilon\phi_{ik}^m(j)\partial D/\partial r_k^m(j)$. Summing over all adjacent nodes k, then, we find that,

$$\begin{aligned} \frac{\partial D}{\partial r_i^m(j)} &= \sum_{k\in\mathcal{N}} \phi_{ik}^m(j)\Big[D'_{ik}(t_i^m(j)\phi_{ik}^m(j)) + \frac{\partial D}{\partial r_k^m(j)}\Big] \\ &= \sum_{k\in\mathcal{N}} \phi_{ik}^m(j)\delta_{ik}^m(j) \end{aligned} \tag{7}$$

Here, $\delta_{ik}^m(j) = D'_{ik}(t_i^m(j)\phi_{ik}^m(j)) + \frac{\partial D}{\partial r_k^m(j)}$ is called the marginal delay of link (i,k) with respect to receiver j. (7) asserts that the marginal delay of a node is the convex sum of the marginal delays of its outgoing links with respect to the same destination. By the definition of $\boldsymbol{\phi}$, we can see that $\partial D/\partial r_j^m(j) = 0$, since $\phi_{jk}^m(j) = 0$, i.e., no traffic of receiver j needs to be routed anymore once it arrives to the destination.

Next consider $\partial D/\partial\phi_{ik}^m(j)$. An increment ϵ in $\phi_{ik}^m(j)$ causes an increment $\epsilon t_i^m(j)$ in the portion of $t_i^m(j)$ flowing on link (i,k). If $k \neq j$, this causes an addition $\epsilon t_i^m(j)$ to the traffic at k destined for j. Thus for $(i,k) \in \mathcal{L}$, $i \neq j$,

$$\begin{aligned} \frac{\partial D}{\partial\phi_{ik}^m(j)} &= t_i^m(j)\Big[D'_{ik}(t_i^m(j)\phi_{ik}^m(j)) + \frac{\partial D}{\partial r_k^m(j)}\Big] \\ &= t_i^m(j)\delta_{ik}^m(j) \end{aligned} \tag{8}$$

To summarize above discussions, we have the following theorems.

Theorem 1: Let a network have inputs $\boldsymbol{r}$ and routing variables $\boldsymbol{\phi}$, and let each marginal delay $dD_{ik}(f_{ik})/df_{ik}$ be continuous in f_{ik}, $(i,k) \in \mathcal{L}$. Then the set of equations (7), $i \neq j$, has a unique (and correct) set of solutions for $\partial D/\partial r_i^m(j)$. Furthermore, (16) is valid and both $\partial D/\partial r_i^m(j)$ and $\partial D/\partial\phi_{ik}^m(j)$ for $i \neq j$, $(i,k) \in \mathcal{L}$ are continuous in $\boldsymbol{r}$ and $\boldsymbol{\phi}$.

Theorem 2: Assume that D_{ik} is convex and continuously differentiable for f_{ik}. let ψ be the set of $\boldsymbol{\phi}$, the necessary condition for $\boldsymbol{\phi}$ to minimize D over ψ is

$$\frac{\partial D}{\partial\phi_{ik}^m(j)} \begin{cases} = \min_l \partial D/\partial\phi_{il}^m(j) & \text{if } \phi_{ik}^m > 0 \\ \geq \min_l \partial D/\partial\phi_{il}^m(j) & \text{if } \phi_{ik}^m = 0 \end{cases} \tag{9}$$

and the sufficient condition for $\boldsymbol{\phi}$ to minimize D over ψ is

$$\delta_{ik}^m(j) \geq \frac{\partial D}{\partial r_i^m(j)}, \forall i \neq j, (i,k) \in \mathcal{L}, \forall m \in \mathcal{M} \tag{10}$$

The necessary condition (9) in Theorem 2 states that within session m, at node i, for a given receiver j, all links (i,k) that have any portion of flow $t_i^m(j)$ routed through ($\phi_{ik}^m(j) > 0$) must achieve the same minimum marginal delay with respect to j, and that this minimum marginal delay must be less than or equal to the same marginal delays of the links with no flow routed ($\phi_{ik}^m(j) = 0$).

The sufficient condition (10) states that within session m, at node i, for a given receiver j, the marginal delay of all links (i,k) with respect to j must be greater than or equal to the marginal delay of node i.

B. Minimum Delay Routing in Overlay Multicast

In the setting of overlay network, we need to redefine the link set $\mathcal{L}$ since each link $(i,k) \in \mathcal{L}$ is actually a unicast route going through a set of physical links. Let $\mathcal{Z}$ be the set of physical links encompassed by the overlay network $\mathcal{L}$, we define function $n_z(i,k)$. $n_z(i,k) = 1$ if link (i,k) goes through the physical link $z \in \mathcal{Z}$, and 0 otherwise. The capacity constraint (4) should be rephrased as

$$f_z = \sum_{(i,k)\in\mathcal{L}} n_z(i,k) \sum_{m\in\mathcal{M}} f_{ik}^m \leq C_z, \forall z \in \mathcal{Z} \tag{11}$$

Also the delay of link (i,k) is the aggregate delay of all physical links it goes through. Therefore,

$$D_{ik}(f_{ik}) = \sum_{z\in\mathcal{Z}} n_z(i,k) D_z(f_z) \tag{12}$$

Then our goal is formalized into the following problem[1].

$$\begin{aligned} \textbf{O:}\quad & \textbf{minimize} \quad D = \sum_{(i,k)\in\mathcal{L}} D_{ik}(f_{ik}) \\ & \textbf{subject to} \quad (1),(2)(\textbf{flow constraint}) \\ & \qquad\qquad\ (3)(\textbf{union of flow constraint}) \\ & \qquad\qquad\ (11)(\textbf{capacity constraint}) \end{aligned}$$

Now let us first derive partial derivative of delay D to an input variable $r_i^m(j)$.

$$\frac{\partial D}{\partial r_i^m(j)} = \sum_{(k,l)\in\mathcal{L}} \sum_{z\in\mathcal{Z}} n_z(k,l) \frac{\partial D_z(f_z)}{\partial r_i^m(j)} \tag{13}$$

where

$$\frac{\partial D_z(f_z)}{\partial r_i^m(j)} = \tag{14}$$
$$\sum_{(k,l)\in\mathcal{L}} n_z(k,l) \frac{dD_z(f_z)}{df_z} \frac{df_{kl}^m}{d(t_k^m(j)\phi_{kl}^m(j))} \frac{d(t_k^m(j)\phi_{kl}^m(j))}{dr_i^m(j)}$$

Here, $dD_z(f_z)/df_z$ is the marginal delay of the physical link z, the definition of $df_{kl}^m/d(t_k^m(j)\phi_{kl}^m(j))$ can be found at (6), and

$$\frac{d(t_k^m(j)\phi_{kl}^m(j))}{dr_i^m(j)} = \sum_{\text{all paths } P \text{ from } i \text{ to } k} \phi_{kl}^m(j) \prod_{(n,p)\in P} \phi_{np}^m(j) \tag{15}$$

Eq. (15) means that if there is a small increment ϵ on the input $r_i^m(j)$, the corresponding increment on link (k,l) will be $\epsilon \cdot d(t_k^m(j)\phi_{kl}^m(j))/dr_i^m(j)$.

Summarizing over Eq. (14) and (15), we find out that Eq. (13) can be simplified into the following recursive form:

$$\frac{\partial D}{\partial r_i^m(j)} = \sum_{k\in\mathcal{N}} \phi_{ik}^m(j) \Big[\frac{\partial D_{ik}(f_{ik})}{\partial r_i^m(j)} + \frac{\partial D}{\partial r_k^m(j)} \Big]$$

which is similar to (7).

Following the same way, we derive the partial derivative of D to routing variable $\phi_{ik}^m(j)$ as follows.

$$\frac{\partial D}{\partial \phi_{ik}^m(j)} = t_i^m(j) \Big[\frac{\partial D_{ik}(f_{ik})}{\partial r_i^m(j)} + \frac{\partial D}{\partial r_k^m(j)} \Big]$$

It can be easily verified that within the overlay network setting, Theorem 1 still holds. From the definition of D_{ik} in Eq. (12), we can see that if the physical link delay $D_z(f_z)$ is convex and continuously differentiable, D_{ik} is also convex and continuously differentiable. Therefore, we are able to reach the following conclusion, which is similar to Theorem 2.

[1]In definitions (11) and (12), we can see that the physical link z can be either unidirectional or bidirectional.

Corollary 1: Assume D_{ik} is convex and continuously differentiable for f_{ik}, let ψ be the set of $\boldsymbol{\phi}$, the necessary condition for $\boldsymbol{\phi}$ to minimize D over ψ is

$$\frac{\partial D}{\partial \phi_{ik}^m(j)} \begin{cases} = \min_l \partial D/\partial \phi_{il}^m(j) & \text{if } \phi_{ik}^m > 0 \\ \geq \min_l \partial D/\partial \phi_{il}^m(j) & \text{if } \phi_{ik}^m = 0 \end{cases} \tag{16}$$

and the sufficient condition for $\boldsymbol{\phi}$ to minimize D over ψ is

$$\frac{\partial D_{ik}(f_{ik})}{\partial r_i^m(j)} + \frac{\partial D}{\partial r_k^m(j)} \geq \frac{\partial D}{\partial r_i^m(j)} \tag{17}$$
$$\forall i \neq j, (i,k) \in \mathcal{L}, \forall m \in \mathcal{M}$$

C. Maximum lifetime Routing in Multihop Wireless Network

In multi-hop wireless network such as sensor network, data is sent through wireless link, which consumes limited battery energy of both sender node and receiver node. Energy-efficient routing thus becomes an important issue. Our goal here is to maximize the lifetime of the network, i.e., the duration in which all nodes are up until one of them is drained of energy.

We define E_i the energy reserve at node i. Let p_i^r (J/bit) be the power consumption at node i, when it receives one unit of data, and p_{ik}^t (J/bit) be the power consumption when one unit of data is sent from i over link (i,k). Based on the first order radio model, we have the following.

$$p_i^r = a \tag{18}$$
$$p_{ik}^t = a + b \cdot (d_{ik})^\theta \tag{19}$$

Here, a is a distance-independent constant that represents the energy consumption to run the transmitter or receiver circuitry, and b is the coefficient of the distance-dependent term that represents the transmit amplifier. d_{ik} is the distance from node i to k. The exponent θ is determined from field measurements, which is typically a constant between 2 and 4. The power consumption ratio (J/s) of node i is

$$p_i = \sum_{k\in\mathcal{N}} \big[f_{ik} \cdot p_{ik}^t + f_{ki} \cdot p^r \big], \forall i \in \mathcal{N} \tag{20}$$

Now it is clear that the lifetime of node i is

$$T_i = \frac{E_i}{p_i} \tag{21}$$

Our target is to maximize the minimum lifetime of all nodes, i.e., the duration that all nodes within the network are up. Associating T_i with a utility U_i, this goal can be

formalized as to maximize the aggregate utility of all nodes as follows.

$$\begin{aligned}\textbf{U:}\quad & \textbf{maximize} && U=\sum_{i\in\mathcal{N}}U_i=\sum_{i\in\mathcal{N}}\frac{T_i^{1-\gamma}}{1-\gamma},\gamma\to\infty\\ & \textbf{subject to} && (1),(2)(\textbf{flow constraint})\\ & && (3)(\textbf{union of flow constraint})\\ & && (4)(\textbf{capacity constraint})\\ & && (20),(21)(\textbf{power constraint})\end{aligned}$$

Here γ can be made an arbitrarily large number to infinitely approximate the optimal value.

We first consider $\partial U/\partial r_i^m(j)$, the *marginal utility* on node i with respect to receiver j. Assume that there is a small increment ϵ on the input traffic $r_i^m(j)$. Then $\epsilon\phi_{ik}^m(j)$ from this new incoming traffic will flow over wireless link (i,k). This will cause an increment power consumption on node i,

$$\epsilon\phi_{ik}^m(j)p_{ik}^t\frac{df_{ik}^m}{d(t_i^m(j)\phi_{ik}^m(j))}$$

in order to send out the incremented traffic. The definition of $df_{ik}^m/d(t_i^m(j)\phi_{ik}^m(j))$ can be found at Eq. (6). And the consequent utility change of node i is

$$\epsilon\phi_{ik}^m(j)U_i'(p_i)p_{ik}^t\frac{df_{ik}^m}{d(t_i^m(j)\phi_{ik}^m(j))}$$

Similarly, on the receiver side, the utility change of node k is

$$\epsilon\phi_{ik}^m(j)U_k'(p_k)p_k^r\frac{df_{ik}^m}{d(t_i^m(j)\phi_{ik}^m(j))}$$

If node k is not the destination node, then the increment $\epsilon\phi_{ik}^m(j)$ of extra traffic at node k will cause the same utility change onward as a result of the increment $\epsilon\phi_{ik}^m(j)$ of input traffic at node k. To first order this utility change will be $\epsilon\phi_{ik}^m(j)\partial U/\partial r_k(j)$. Summing over all adjacent nodes k, then, we find that,

$$\begin{aligned}\frac{\partial U}{\partial r_i^m(j)} &= \sum_{k\in\mathcal{N}}\phi_{ik}^m(j)\Big[\frac{\partial U}{\partial r_k^m(j)}+\\ &\quad (p_{ik}^tU_i'(p_i)+p_k^rU_k'(p_k))\frac{df_{ik}^m}{d(t_i^m(j)\phi_{ik}^m(j))}\Big]\\ &= \sum_{k\in\mathcal{N}}\phi_{ik}^m(j)\Big[U_{ik}'(t_i^m(j)\phi_{ik}^m(j))+\frac{\partial U}{\partial r_k^m(j)}\Big] \qquad (22)\end{aligned}$$

where $U_{ik}'(t_i^m(j)\phi_{ik}^m(j)) = (p_{ik}^tU_i'(p_i)+p_k^rU_k'(p_k))\cdot\frac{df_{ik}^m}{d(t_i^m(j)\phi_{ik}^m(j))}$ is called the marginal utility on link (i,k), and $U_{ik}'(t_i^m(j)\phi_{ik}^m(j))+\frac{\partial U}{\partial r_k^m(j)}$ is called the marginal utility of link (i,k) with respect to receiver j.

(7) asserts that the marginal utility of a node is the convex sum of the marginal utilities of its outgoing links with respect to the same receiver. By the definition of $\boldsymbol{\phi}$, we can see that $\partial U/\partial r_j^m(j)=0$, since $\phi_{jk}^m(j)=0$, i.e., no traffic of receiver j needs to be routed anymore once it arrives to the destination.

Next we consider $\partial U/\partial\phi_{ik}^m(j)$. An increment ϵ in $\phi_{ik}^m(j)$ causes an increment $\epsilon t_i^m(j)$ in the portion of $t_i^m(j)$ flowing on link (i,k). If $k\neq j$, this causes an addition $\epsilon t_i^m(j)$ to the traffic at k destined for j. Thus for $(i,k)\in\mathcal{L}$, $i\neq j$,

$$\frac{\partial U}{\partial\phi_{ik}^m(j)}=t_i^m(j)\Big[U_{ik}'(t_i^m(j)\phi_{ik}^m(j))+\frac{\partial U}{\partial r_k^m(j)}\Big] \qquad (23)$$

We are able to prove the following corollaries similar to Theorem 1 and 2.

Corollary 2: Let a wireless network have inputs $\boldsymbol{r}$ and routing variables $\boldsymbol{\phi}$, and let each marginal utility $U_i'(p_i)$ be continuous in p_i, $i\in\mathcal{N}$. Then the set of equations (22), $i\neq j$, has a unique (and correct) set of solutions for $\partial U/\partial r_i^m(j)$. Furthermore, (23) is valid and both $\partial U/\partial r_i^m(j)$ and $\partial U/\partial\phi_{ik}^m(j)$ for $i\neq j$, $(i,k)\in\mathcal{L}$ are continuous in $\boldsymbol{r}$ and $\boldsymbol{\phi}$.

Corollary 3: Assume that U_i is concave and continuously differentiable for p_i. let ψ be the set of $\boldsymbol{\phi}$, the necessary condition for $\boldsymbol{\phi}$ to maximize U over ψ is

$$\frac{\partial U}{\partial\phi_{ik}^m(j)}\begin{cases}=\max_l\partial U/\partial\phi_{il}^m(j) & \text{if } \phi_{ik}^m>0\\ \leq\max_l\partial U/\partial\phi_{il}^m(j) & \text{if } \phi_{ik}^m=0\end{cases} \qquad (24)$$

and the sufficient condition for $\boldsymbol{\phi}$ to maximize U over ψ is

$$\begin{aligned}U_{ik}'(t_i^m(j)\phi_{ik}^m(j))+\frac{\partial U}{\partial r_k^m(j)}\leq\frac{\partial U}{\partial r_i^m(j)} \qquad (25)\\ \forall i\neq j,(i,k)\in\mathcal{L},\forall m\in\mathcal{M}\end{aligned}$$

Note that since U is decreasing and concave in f_{ik} while D is increasing and convex in f_{ik}, the optimality conditions in Corollary 3 are exactly opposite to the ones in Theorem 2. Also note that by (21) and the definition of U, we can see that U is concave as long as Eq. (20) is convex in f_{ik}. Therefore, as long as it is a convex function of f_{ik}, the power consumption model does not need to follow the definition in Eq. (19) and (18).

IV. Distributed Routing Algorithm

By understanding the optimality conditions (general model discussed in Sec. III-A) to multicast routing, the design philosophy of our routing scheme should now be clear. The algorithm works in an iterative fashion. In each iteration, for each session m, each node i and

a given receiver j, i must incrementally increase the fraction of traffic on link (i,k) (by increasing $\phi_{ik}^m(j)$) whose marginal delay $\delta_{ik}^m(j)$ is small, and do the reverse for those links whose marginal delay is big, until the marginal delays of all links carrying traffic are equal. When this condition is met for all nodes regarding all receivers within all sessions, the entire system reaches the optimal point.

Therefore, for each session m, each node i, each iteration involves two steps: (1) the calculation of marginal delay $D'_{ik}(t_i^m(j)\phi_{ik}^m(j))$ for each outgoing link (i,k), and each of its downstream neighbors k's marginal delay $\partial D/\partial r_k^m(j)$; (2) the adjustment of routing variables $\phi_{ik}^m(j)$ based on the values of $D'_{ik}(t_i^m(j)\phi_{ik}^m(j))$ and $\partial D/\partial r_k^m(j)$. We will elaborate them in details as follows.

Sec. IV-A introduces how the calculation and update of marginal delays $D'_{ik}(t_i^m(j)\phi_{ik}^m(j))$ and $\partial D/\partial r_k^m(j)$ are executed. Sec. IV-B discusses how to maintain loop-free routing. Sec. IV-C formally presents the algorithm, whose optimal property is analyzed in Sec. IV-D.

Finally, in Sec. IV-E and IV-F, we discuss how the algorithm should be adjusted in the setting of minimum-delay routing in overlay network and maximum-lifetime routing in wireless network.

A. Calculation of Marginal Delays

We first see how each node i calculates its marginal delay $\partial D/\partial r_i^m(j)$, with respect to receiver j. In order to do so, based on Eq. (7), i needs to know $\delta_{ik}(j) = D'_{ik}(t_i^m(j)\phi_{ik}^m(j)) + \partial D/\partial r_k^m(j)$, the marginal delays of all its outgoing links regarding receiver j. In Sec. III-A, we have discussed how to calculate $D'_{ik}(t_i^m(j)\phi_{ik}^m(j))$, and $\partial D/\partial r_k^m(j)$ is the marginal delay of i's downstream neighbor k. Now it is clear that $\partial D/\partial r_i(j)$ should be calculated in a recursive way. Starting from receiver, $\partial D/\partial r_j^m(j) = 0$ based on definition. j then sends the values of $\partial D/\partial r_j^m(j)$ to its upstream neighbor, say k. Upon receiving the updates, node k can calculates $D'_{ik}(t_i^m(j)\phi_{ik}^m(j))$ as described above, then acquire $\partial D/\partial r_k^m(j)$. Then, k repeats the same procedure to its upstream neighbor, until node i is reached.

B. Loop-free Routing

From the above calculation, we can see that among all nodes carrying traffic of session m, their marginal delays follow a partial ordering. Each receiver j has the lowest marginal delay, which is 0. Its upstream neighbors have higher marginal delays, whose own upstream neighbors have even higher marginal delays. Therefore, the recursive procedure of node marginal delay calculation is free of deadlock if and only if such a partial ordering is maintained, i.e., the routing variable set $\boldsymbol{\phi}$ is loop free.

In order to achieve loop-free routing, for each node i, with respect to receiver j, we introduce a set $B_{i,\boldsymbol{\phi}}^m(j)$ of blocked nodes k for which $\phi_{ik}^m(j) = 0$ and the algorithm is not permitted to increase $\phi_{ik}^m(j)$ from 0. $k \in B_{i,\boldsymbol{\phi}}^m(j)$ if one of the following conditions is met.

1) $(i,k) \notin \mathcal{L}$, i.e., k is not the neighbor of i.
2) $\phi_{ik}^m(j) = 0$ and $\partial D/\partial r_i^m(j) \le \partial D/\partial r_k^m(j)$, i.e., the marginal delay of k is already greater than or equal to the marginal delay of i.
3) $\phi_{ik}^m(j) = 0$ and $\exists (l,p) \in \mathcal{L}$ such that (a) $l = k$ or l is downstream to k with respect to receiver j; (b) $\phi_{lp}^m(j) > 0$, and $\partial D/\partial r_l^m(j) \le \partial D/\partial r_p^m(j)$, i.e., (l,p) is an improper link.

An example illustrating improper link is shown in Fig. 2. The solid line indicates that there is traffic on this link, and the dotted line indicates otherwise. Here node 4 is a receiver of session m. The partial ordering of their marginal delays are $1 \to 2 \to 3 \to 4$, which the traffic from node 3 to 1 is against. Node 2, if unaware of the existence of such an improper link downstream, might make a loop by moving some of its outgoing traffic to node 3. To prevent this case from happening, node 3 only needs to raise a flag when updating its marginal delay to its upstream nodes 2 and 3. Upon receiving such a notification, nodes 2 and 3 can include node 3 into their blocking sets.

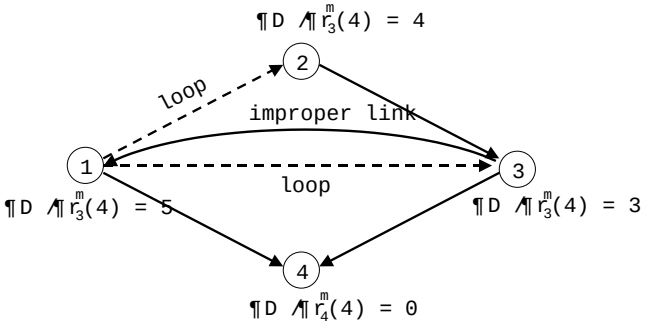


Fig. 2. Illustration of Improper Link

C. Algorithm

Now we are ready to formalize our algorithm. We use $\boldsymbol{\phi}^{(k)}$ to represent the routing variable set at the iteration k. $\Delta\boldsymbol{\phi}^{(k)}$ is the changes made to $\boldsymbol{\phi}^{(k)}$ during the iteration k. Apparently, $\boldsymbol{\phi}^{(k+1)} = \boldsymbol{\phi}^{(k)} + \Delta\boldsymbol{\phi}^{(k)}$. Also for node i,

- $\boldsymbol{\phi}_i^m(j) = (\phi_{i1}^m(j), \ldots, \phi_{in}^m(j))^T$ is the vector of its routing variable regarding receiver j and session m.
- $\Delta\boldsymbol{\phi}_i^m(j) = (\Delta\phi_{i1}^m(j), \ldots, \Delta\phi_{in}^m(j))^T$ is the vector of changes to $\boldsymbol{\phi}_i^m(j)$.

- $\boldsymbol{\delta}_i^m(j) = (\delta_{i1}^m(j), \ldots, \delta_{in}^m(j))^T$ is the vector of marginal delays of all i's neighbors.

At iteration k, node i operates according to the following steps.

1) For each session m, calculate link marginal delay $D'_{ik}(t_i^m(j)\phi_{ik}^m(j))$ for each of its outgoing links (i,k), get updates of marginal delays $\partial D/\partial r_k^m(j)$ from each of its downstream neighbors k, then calculate $\delta_{ik}^m(j) = D'_{ik}(t_i^m(j)\phi_{ik}^m(j)) + \partial D/\partial r_k^m(j)$.
2) Calculate its own marginal delay $\partial D/\partial r_i^m(j)$ according to Eq. (7), and send it to all its upstream neighbors.
3) Calculate $\boldsymbol{\phi}_i^m(j)^{(k)}$ by solving the problem

$$\begin{aligned} \textbf{minimize} \quad & \boldsymbol{\delta}_i^m(j)^T \Delta\boldsymbol{\phi}_i^m(j) + \frac{t_i^m(j)}{2\alpha} \cdot \\ & (\Delta\boldsymbol{\phi}_i^m(j)^{(k)})^T \boldsymbol{M}_i^m(j)^{(k)} \Delta\boldsymbol{\phi}_i^m(j)^{(k)} \\ \textbf{subject to} \quad & \boldsymbol{\phi}_i^m(j)^{(k)} + \Delta\boldsymbol{\phi}_i^m(j)^{(k)} \geq 0, \\ & \sum_{l\in\mathcal{N}} \Delta\phi_{il}^m(j)^{(k)} = 0, \Delta\phi_{il}^m(j)^{(k)} = 0, \\ & \forall l \in B_{i,\boldsymbol{\phi}^{(k)}}^m(j) \end{aligned} \tag{26}$$

   where $\alpha > 0$ is some positive stepsize, and matrix $\boldsymbol{M}_i^m(j)^{(k)}$ is some symmetric matrix which is positive definite on the subspace $\{\Delta\boldsymbol{\phi}_i^m(j) \mid \sum_{l\in\mathcal{N}} \Delta\phi_{il}^m(j) = 0\}$.
4) Adjust routing variables

$$\begin{aligned} & \boldsymbol{\phi}_i^m(j)^{(k+1)} = \boldsymbol{\phi}_i^m(j)^{(k)} + \Delta\boldsymbol{\phi}_i^m(j)^{(k)} \\ & \forall i \in \mathcal{N} - \{j\}, \forall m \in \mathcal{M} \end{aligned}$$

Note that in problem (26), $M_i^m(j)^{(k)}$ can be any positive definite matrix, and any solution $\Delta\boldsymbol{\phi}_i^m(j)$ to this problem will allocate more traffic on the link with the minimum marginal delay, and decrease traffic on other links. If we implement $M_i^m(j)^{(k)}$ as the identity matrix, the solution to $\Delta\boldsymbol{\phi}^m(j)^{(k)}$ boils down to

$$\begin{aligned} & \Delta\phi_{il}^m(j)^{(k)} \\ = & \begin{cases} 0 & \text{if } l \in B_{i,\boldsymbol{\phi}^{(k)}}^m(j) \\ -\min\{\phi_{il}^m(j)^{(k)}, & \\ \frac{\alpha(\delta_{il}^m(j) - \delta_{\min}^m(j))}{t_i^m(j)}\} & \text{if } \delta_{il}^m(j) \neq \delta_{\min}^m(j) \\ \sum_{\delta_{ip}^m(j) \neq \delta_{\min}^m(j)} \Delta\phi_{ip}^m(j)^{(k)} & \text{if } \delta_{il}^m(j) = \delta_{\min}^m(j) \end{cases} \end{aligned}$$

where $\delta_{\min}^m(j) = \min_{p\notin B_{i,\boldsymbol{\phi}^{(k)}}^m(j)} \delta_{ip}(j)$.

This algorithm increase the fraction of traffic on the link with the minimum marginal delay, and reduces the fraction of other links. The amount of reduction on link (i,l), given by $\Delta\phi_{il}^m(j)^{(k)}$, is proportional to $\delta_{il}^m(j) - \delta_{\min}^m(j)$, the difference of marginal delays between (i,l) itself and the link with the minimum marginal delay. It is further restricted that $\Delta\phi_{il}^m(j)^{(k)} \leq \phi_{il}^m(j)^{(k)}$, i.e., $\Delta\phi_{il}^m(j)^{(k)}$ should not turn $\phi_{il}^m(j)^{(k)}$ to negative. The amount of reduction is also inversely proportional to $t_i^m(j)$, since the change in link traffic is related to $\Delta\phi_{il}^m(j)^{(k)} t_i^m(j)$. When $t_i^m(j)$ is small, $\Delta\phi_{il}^m(j)^{(k)}$ can be changed by a large amount without greatly affecting the marginal delays. Finally, the change depends on the stepsize α. As shown later in Theorem 3, convergence can be guaranteed if α is small enough. As α increases, the speed of convergence increases but the danger of no convergence also increases.

We can implement $\boldsymbol{M}_i^m(j)^{(k)}$ differently to further improve convergence speed. For example, Bertsekas et al.[4] choose to set $\boldsymbol{M}_i^m(j)^{(k)}$ as a diagonal matrix where the element at the lth row and lth column is the second derivative[2] of delay D to routing variable $\phi_{il}^m(j)$, i.e., $\partial^2 D/(\partial\phi_{il}^m(j))^2$.

D. Analysis

The following lemma shows some of the properties of our algorithm.

Lemma 2:
(a) If $\boldsymbol{\phi}^{(k)}$ is loop-free, then $\boldsymbol{\phi}^{(k+1)}$ is loop-free.
(b) If $\boldsymbol{\phi}^{(k)}$ is loop-free and $\Delta\boldsymbol{\phi}^{(k)} = 0$ solves problem defined in step (3) of the algorithm, then $\boldsymbol{\phi}^{(k)}$ is optimal.
(c) If $\boldsymbol{\phi}^{(k)}$ is optimal, then $\boldsymbol{\phi}^{(k+1)}$ is also optimal.
(d) If $\Delta\boldsymbol{\phi}^{(k)} \neq 0$ for some i for which $t_i^m(j) > 0$, then there exists a positive scalar η_k such that

$$D(\boldsymbol{\phi}^{(k)} + \eta\Delta\boldsymbol{\phi}^{(k)}) < D(\boldsymbol{\phi}^{(k)}), \forall \eta \in (0, \eta_k]$$

The following theorem shows the main convergence result.

Theorem 3: Let the initial routing $\boldsymbol{\phi}^{(0)}$ be loop-free and satisfy $D(\boldsymbol{\phi}^{(0)}) \leq D_0$ where D_0 is some scalar. Assume also that there exist two positive scalars λ, Λ such that for each session m, each node i, and each receiver j, the sequences of matrices $\{\boldsymbol{M}_i^m(j)^{(k)}\}$ satisfy the following two conditions.

(a) The absolute value of each element of $\boldsymbol{M}_i^m(j)^{(k)}$ is bounded above by Λ.

(b) There holds

$$\lambda|\boldsymbol{v}_i|^2 \leq \boldsymbol{v}_i^T \boldsymbol{M}_i^m(j)^{(k)} \boldsymbol{v}_i$$

for all $\boldsymbol{v}_i$ such that $\sum_{l\notin B_{i,\boldsymbol{\phi}^{(k)}}^m(j)} v_{il} = 0$.

Then there exists a positive scalar $\overline{\alpha}$ (depending on D_0, λ, and Λ) such that for all $\alpha \in (0, \overline{\alpha}]$ and $k =$

[2]In fact, since $\partial^2 D/(\partial\phi_{il}^m(j))^2$ is difficult to compute, this element is usually set to be its upper bound.

$0, 1, \ldots$, the sequence $\{\boldsymbol{\phi}^{(k)}\}$ generated by the algorithm satisfies

$$D(\boldsymbol{\phi}^{(k+1)}) \leq D(\boldsymbol{\phi}^{(k)})$$
$$\lim_{k\to\infty} D(\boldsymbol{\phi}^{(k+1)}) = \min_{\boldsymbol{\phi}\in\psi} D(\boldsymbol{\phi})$$

Furthermore, every limit point of $\{\boldsymbol{\phi}^{(k)}\}$ is an optimal solution to problem defined in step (3) of the algorithm.

E. Distributed Algorithm for Minimum-Delay Routing in Overlay Network

From the optimality conditions (16) and (17) derived in Sec. III-B, we see that the algorithm presented in this section can be directly applied into the setting of overlay network.

The only exception is that an overlay link (i, k) is a unicast route containing several physical links. Hence its delay (marginal delay) is the aggregate delay (marginal delay) of all these links. In order to calculate the marginal delay of (i, k), it is impractical, if not at all impossible, to calculate the marginal delays of all physical links on its route, then add them up.

Instead, we can treat the delay function D_{ik} as a black box and monitor the change of its output (end-to-end delay of overlay link (i, k)) reacting to the change of its input ($r_i^m(j)$), then estimate its derivative (marginal delay $\partial D_{ik}/\partial r_i^m(j)$). Such a technique is called perturbation analysis[13], which we will briefly mention in Sec. V-A.

F. Distributed Algorithm for Maximum-Lifetime Routing in Wireless Network

The algorithm for the general model can be applied into the setting of wireless network with the following changes.

First, the calculation of link marginal utility $U'_{ik}(t_i^m(j)\phi_{ik}^m(j)) = (p_{ik}^t U_i'(p_i) + p_k^r U_k'(p_k))\frac{df_{ik}^m}{d(t_i^m(j)\phi_{ik}^m(j))}$ requires cooperation of both sender i and receiver k, since sending data over the wireless link (i, k) requires power consumption of both nodes. Node i can calculate $df_{kl}^m/d(t_k^m(j)\phi_{kl}^m(j))$ based on Eq. (6). i is also responsible to calculate the term $p_{ik}^t U_i'(p_i)$. $U_i'(p_i)$ can be derived based on the definition of U, if the energy reserve E_i and power consumption ratio p_i are known. p_{ik}^t can be calculated based on Eq. (19), if constants a, b, θ, and node distance d_{ik} are known beforehand. Node k is responsible to calculate the term $p_k^r U_k'(p_k)$. $U_k'(c_k)$ can be calculated the same way as $U_i'(c_i)$. p_k^r can be calculated based on Eq. (18). After calculation, k can send the value of $p_k^r U_k'(p_k)$ to node i, which in turn acquires $U'_{ik}(t_i^m(j)\phi_{ik}^m(j))$.

Second, by the optimality conditions (16) and (17) derived in Sec. III-B, and the fact that utility function U is decreasing and concave in f_{ik}, the algorithm should do the following. In each iteration, for each session m, each node i and a given receiver j, i must incrementally increase the fraction of traffic on link (i, k) whose marginal utility is great, and do the reverse for those links whose marginal utility is small, until the marginal utilities of all links carrying traffic are equal. Consequently, in the formal algorithm presented in Sec. IV-C, problem (26) should be redefined to:

$$\textbf{maximize} \quad \boldsymbol{\delta}_i^m(j)^T \Delta\boldsymbol{\phi}_i^m(j) - \frac{t_i^m(j)}{2\alpha} \cdot (\Delta\boldsymbol{\phi}_i^m(j)^{(k)})^T \boldsymbol{M}_i^m(j)^{(k)} \Delta\boldsymbol{\phi}_i^m(j)^{(k)} \tag{27}$$

subject to the same constraints.

Here, $\boldsymbol{\delta}_i^m(j) = (\delta_{i1}^m(j), \ldots, \delta_{in}^m(j))^T$ is a vector where $\delta_{il}^m(j) = U'_{il}(t_i^m(j)\phi_{il}^m(j)) + \partial U/\partial r_l^m(j)$, the marginal utility of link (i, l) in session m, with respect to receiver j .

V. Practical Issues

A. Measurements of Marginal Delay and Marginal Utility

In the real minimum-delay routing environment, we cannot assume the delay function of a link to be exactly the same as what is defined in Eq. (5). In the setting of overlay network, the end host may not even know the capacity of some physical link its unicast route goes through. Furthermore, the overhead of calculating marginal delays of all physical links and aggregate them to acquire the marginal delay of an overlay link, is unacceptable, if such an operation is not at all impossible to execute.

In [13], a procedure is presented for estimating online marginal packet delays through links with respect to link flows without making the standard assumptions (exponentially distributed packet lengths, Poisson arrival processes). This procedure is based on a technique known as perturbation analysis. No knowledge of network parameters (arrival rates, link capacities) is required. The same technique can be employed in both physical and overlay network environment.

Similarly, in maximum-lifetime wireless routing environment, we can adopt the same approach. During the calculation of marginal utility $U'_{ik}(t_i^m(j)\phi_{ik}^m(j))$, node i or k can estimate its power consumption ratio by directly measuring the amount of data sent and the corresponding energy dissipation during the most recent period, then derive the marginal utility based on Eq. (21) and the definition of U, both of which are predefined

independent of power consumption models of wireless nodes.

B. Messaging Overhead

In each iteration of our algorithm, the destination node of each link needs to update the marginal delay or marginal utility of this link to the source node. Therefore, a total of $|\mathcal{L}|$ messages need to be sent, $|\mathcal{L}|$ being the number of links inside the network. In case there are more than one multicast sessions, the number of messages required can stay unchanged if each node aggregates its marginal delays or marginal utilities regarding all sessions into a single message. Such messaging overhead can be further saved if we piggyback these messages into data/acknowledgement packets.

C. Interference of Wireless Transmission

It is well known that the achievable rates in multi-hop wireless network are not only constrained the capacities of wireless links, but also location-dependent contention and spatial reuse[14], [15]. Given the fact that deriving the optimal achievable rates is NP-hard, [15] gave an approximation algorithm, which is guaranteed to return a packet scheduling solution which is within 67% of the optimal solution. In our case, we can choose to run this scheduling algorithm, then reset the capacity C_{ik} of each wireless link (i,k) to its maximal achievable rate. In this way, we guarantee that our routing solution is always schedulable at the price of suboptimal bandwidth utilization.

As we have argued in the introduction, maximizing throughput is not the most urgent issue, given the current asymmetric situation of bandwidth supply and application demand in wireless network. Rather, we consider the battery energy on wireless node as the most precious resource, hence the lifetime of the entire network, which our algorithm tries to optimize.

VI. Conclusion

This paper presents a general solution for optimal multicast routing. We show that with the aid of network coding, the once intractable optimal multicast routing problem becomes tractable. We further show that this problem can be solved in an entirely distributed fashion by presenting a distributed routing algorithm, which is proved to converge to the point where the value of the objective function is optimized. Our solution can be fit into a variety of networks to achieve different optimization goals, such as minimum delay routing in overlay multicast, and maximum lifetime routing in multi-hop wireless network.

VII. Appendix A: Proof of Lemma 1

Proof: Without loss of generality, let us consider the commodity j in session m. We restate Eq. (2) as

$$t_i^m(j) = r_i^m(j) + \sum_{l \in \mathcal{N}-\{j\}} t_l^m(j)\phi_{li}^m(j), \forall i \in \mathcal{N}, \forall m \in \mathcal{M} \tag{28}$$

since $\phi_{lj}^m(j) = 0$. Summing both sides over i, we have

$$t_j^m(j) = \sum_{i \in \mathcal{N}} r_i^m(j) \tag{29}$$

The physical meaning of (29) is obvious: in a session m, the amount of commodity arrived at node j equals the total amount generated from each of its sources. For the pure purpose of proof, we temporarily define $\phi_{ji}^m(j) = r_i^m(j)/t_j^m(j)$ and substitute it into (28), we have

$$t_i^m(j) = \sum_{l \in \mathcal{N}} t_l^m(j)\phi_{li}^m(j), i \in \mathcal{N}, m \in \mathcal{M} \tag{30}$$

Any solution to (30) and (29) satisfies (28). Let $\hat{\mathbf{\Phi}}^m(j)$ be the $n \times n$ matrix with elements $\phi_{li}^m(j)$. $\hat{\mathbf{\Phi}}^m(j)$ is stochastic, since each element $\phi_{li}^m(j) \geq 0$, and $\sum_{i=1}^n \phi_{li}^m(j) = 1$ ($1 \leq i \leq n$,$m \in \mathcal{M}$). Consequently, (30) is the formula for steady-state probabilities in a Markov chain.

If $\hat{\mathbf{\Phi}}^m(j)$ is irreducible, then (30) has a unique solution. In order to make $\hat{\mathbf{\Phi}}^m(j)$ irreducible, there has to exist a path between any pair i and k, i.e., $\phi_{il}^m(j) > 0, \phi_{lm}^m(j) > 0, \ldots, \phi_{pk}^m(j) > 0$. To prove this, we only need to show that there exists a path from node j to any other node, and a path from any other node to j. For a node i, if $r_i(j) > 0$, then there is a path from i to j. Otherwise, the traffic generated from i will not arrive at j, contradicting (29). Also by the temporary definition of $\phi_{ji}^m(j) = r_i^m(j)/t_j^m(j)$, there is a path from j to i too. In conclusion, if $r_i^m(j) > 0 (i \in \mathcal{N} - \{j\}, m \in \mathcal{M})$, then $\hat{\mathbf{\Phi}}^m(j)$ is irreducible, hence 30 has a unique solution, where $t_i^m(j) > 0 (i \in \mathcal{N} - \{j\}, m \in \mathcal{M})$.

If we remove the jth column and jth row of $\hat{\mathbf{\Phi}}^m(j)$, we acquire a $(n-1) \times (n-1)$ matrix $\mathbf{\Phi}^m(j)$. If we define two row vectors as:

$$\begin{aligned} \boldsymbol{t}^m(j) &= (t_1^m(j), \ldots, t_{j-1}^m(j), t_{j+1}^m(j), \ldots, t_n^m(j)) \\ \boldsymbol{r}^m(j) &= (r_1^m(j), \ldots, r_{j-1}^m(j), r_{j+1}^m(j), \ldots, r_n^m(j)) \end{aligned}$$

then we can restate (28) into the following vector form:

$$\boldsymbol{t}^m(j)(\boldsymbol{I} - \mathbf{\Phi}^m(j)) = \boldsymbol{r}^m(j)$$

Since this equation has a unique solution if $\boldsymbol{r}^m(j) > 0$, $I - \mathbf{\Phi}^m(j)$ must have an inverse. Therefore,

$$\boldsymbol{t}^m(j) = \boldsymbol{r}^m(j)(\boldsymbol{I} - \mathbf{\Phi}^m(j))^{-1} \tag{31}$$

Since $\boldsymbol{t}^m(j)$ is positive when $\boldsymbol{r}^m(j)$ is positive, $\boldsymbol{t}^m(j)$ is nonnegative when $\boldsymbol{r}^m(j)$ is nonnegative. Now we differentiate $\boldsymbol{t}^m(j)$ as a function of $\boldsymbol{r}^m(j)$. Differentiating (31), we get the continuous function of $\boldsymbol{\Phi}^m(j)$,

$$\frac{\partial t_i^m(j)}{\partial r_l^m(j)} = [(I - \boldsymbol{\Phi}^m(j))^{-1}]_{li} \tag{32}$$

Using (32) in (31), we can express the solution to (28) as

$$t_i^m(j) = \sum_{l \in \mathcal{N}-\{j\}} \frac{\partial t_i^m(j)}{\partial r_l^m(j)} r_l^m(j) \tag{33}$$

Now we differentiate $\boldsymbol{t}^m(j)$ as a function of $\boldsymbol{\Phi}^m(j)$. Differentiating (28) with $\phi_{kp}^m(j)$, we get

$$\frac{\partial t_i^m(j)}{\partial \phi_{kp}^m(j)} = \begin{cases} \sum_{l \in \mathcal{N}-\{j\}} \frac{\partial t_l^m(j)}{\partial \phi_{kp}^m(j)} \phi_{li}^m(j) + t_k^m(j) & \text{if } i = p \\ \sum_{l \in \mathcal{N}-\{j\}} \frac{\partial t_l^m(j)}{\partial \phi_{kp}^m(j)} \phi_{li}^m(j) & \text{otherwise} \end{cases} \tag{34}$$

If we fix k and p, and introduce two variables $\alpha_i^m(j)$ and $\beta_i^m(j)$ defined as

$$\begin{aligned} \alpha_i^m(j) &= \frac{\partial t_i^m(j)}{\partial \phi_{kp}^m(j)} \\ \beta_i^m(j) &= \begin{cases} t_k^m(j) & \text{if } i = p \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

(34) becomes

$$\alpha_i^m(j) = \beta_i^m(j) + \sum_{l \in \mathcal{N}-\{j\}} \alpha_i^m(j) \phi_{li}^m(j), i \in \mathcal{N}$$

which has the same set of equations as (28), with $\alpha_i^m(j)$ corresponding to $t_i^m(j)$, and $\beta_i^m(j)$ corresponding to $r_i^m(j)$. Also since $\beta_i^m(j) \geq 0$, we can repeat the same derivation for $t_i^m(j)$ and $r_i^m(j)$ and reach the same conclusion as in (32) and (33):

$$\frac{\partial \alpha_i^m(j)}{\partial \beta_l^m(j)} = \frac{\partial t_i^m(j)}{\partial r_l^m(j)} = [(\boldsymbol{I} - \boldsymbol{\Phi}^m(j))^{-1}]_{li}$$

$$\alpha_i^m(j) = \sum_{l \in \mathcal{N}} \frac{\partial \alpha_i^m(j)}{\partial \beta_l^m(j)} \beta_l^m(j) = \frac{\partial \alpha_i^m(j)}{\partial \beta_p^m(j)} \beta_p^m(j)$$

Substituting $\frac{\partial t_i^m(j)}{\partial \phi_{kp}^m(j)}$ and $t_k^m(j)$ back to the above equation, we have the solution, continuous in $\boldsymbol{\phi}^m(j)$, as

$$\frac{\partial t_i^m(j)}{\partial \phi_{kp}^m(j)} = \frac{\partial t_i^m(j)}{\partial r_p^m(j)} t_k^m(j) \tag{35}$$

■

VIII. Appendix B: Proof of Theorem 1

Proof: Without loss of generality, let us consider the commodity j in session m. Let $b_i^m(j) = \sum_{k \in \mathcal{N}} \phi_{ik}^m(j) D'_{ik}(r_i^m(j) \phi_{ik}^m(j))$. We define two column vectors as:

$$\begin{aligned} \boldsymbol{b}^m(j) &= (b_1^m(j), \ldots, b_{j-1}^m(j), b_{j+1}^m(j), \ldots, b_n^m(j))^T \\ \boldsymbol{\nabla} \cdot D^m(j) &= (\partial D/\partial r_1^m(j), \ldots, \partial D/\partial r_{j-1}^m(j), \\ &\quad \partial D/\partial r_{j+1}^m(j), \ldots, \partial D/\partial r_n^m(j))^T \end{aligned}$$

then we can rewrite (7) into the following vector form:

$$\boldsymbol{\nabla} \cdot D^m(j) = \boldsymbol{b}^m(j) + \boldsymbol{\Phi}^m(j)(\boldsymbol{\nabla} \cdot D^m(j)) \tag{36}$$

We saw in the proof of Lemma 1 that $\boldsymbol{I} - \boldsymbol{\Phi}^m$ has a unique inverse. Thus the unique solution to (36), continuous in $\boldsymbol{\Phi}(j)$, is given by

$$\boldsymbol{\nabla} \cdot D^m(j) = (\boldsymbol{I} - \boldsymbol{\Phi}^m(j))^{-1} \boldsymbol{b}^m(j)$$

Substituting $\sum_{k \in \mathcal{N}} \phi_{ik}^m(j) D'_{ik}(t_i^m(j) \phi_{ik}^m(j))$ back to the above equation, we have

$$\begin{aligned} \frac{\partial D}{\partial r_i^m(j)} &= \sum_{l \in \mathcal{N}} \frac{\partial t_l^m(j)}{\partial r_i^m(j)} \sum_{p \in \mathcal{N}} \phi_{lp}^m(j) D'_{lp}(t_l^m(j) \phi_{lp}^m(j)) \\ &= \sum_{(l,p) \in \mathcal{L}} \phi_{lp}^m(j) \frac{\partial t_l^m(j)}{\partial r_i^m(j)} D'_{lp}(t_l^m(j) \phi_{lp}^m(j)) \end{aligned} \tag{37}$$

Finally, differentiating D with $\phi_{ik}^m(j)$ using (3)), we have

$$\begin{aligned} \frac{\partial D}{\partial \phi_{ik}^m(j)} &= \sum_{(l,p) \in \mathcal{L}} D'_{lp}(t_l^m(j) \phi_{lp}^m(j)) \phi_{lp}^m(j) \frac{\partial t_l^m(j)}{\partial \phi_{ik}^m(j)} \\ &\quad + D'_{ik}(t_i^m(j) \phi_{ik}^m(j)) t_i^m(j) \end{aligned}$$

Also from the proof of Lemma 1, we have

$$\begin{aligned} \frac{\partial D}{\partial \phi_{ik}^m(j)} &= t_i^m(j) \sum_{(l,p) \in \mathcal{L}} D'_{lp}(t_l^m(j) \phi_{lp}^m(j)) \phi_{lp}^m(j) \frac{\partial t_l^m(j)}{\partial r_k^m(j)} \\ &\quad + t_i^m(j) D'_{ik}(t_i^m(j) \phi_{ik}^m(j)) \end{aligned}$$

By (37), we have

$$\frac{\partial D}{\partial \phi_{ik}^m(j)} = t_i^m(j) \Big[\frac{\partial D}{\partial r_k^m(j)} + D'_{ik}(t_i^m(j) \phi_{ik}^m(j)) \Big]$$

which is the same as (16). Now we can conclude that (16) is continuous in $\boldsymbol{\phi}(j)$ given the continuity of $t_i(j)$ and $\frac{\partial D}{\partial r_i(j)}$. ■

APPENDIX C: PROOF OF THEOREM 2

Proof: First we show that (9) is a necessary condition to minimize D by assuming that $\boldsymbol{\phi}$ does not satisfy (9). This means that there exists a session m, and nodes i,j,k, and p such that

$$\phi_{ik}(j) > 0, \frac{\partial D(\boldsymbol{\phi})}{\partial \phi_{ik}(j)} > \frac{\partial D(\boldsymbol{\phi})}{\partial \phi_{ip}(j)}$$

Since these derivatives are continuous, a sufficiently small decrease in $\phi_{ik}^m(j)$ and corresponding increase in $\phi_{ip}(j)$ will decrease D, contradicting the fact that $\boldsymbol{\phi}$ does not minimize D.

Next we show that (10) is a sufficient condition to minimize D. Suppose that $\boldsymbol{\phi}$ satisfies (10) and has node flows $\boldsymbol{t}$ and link flows $\boldsymbol{f}$. Let $\boldsymbol{\phi}^*$ be any other set of routing variables with node flows $\boldsymbol{t}^*$ and link flows $\boldsymbol{f}^*$. Define

$$f_{ik}(\lambda) = (1-\lambda) f_{ik} + \lambda f_{ik}^* \quad (38)$$

$$D(\lambda) = \sum_{(i,k)\in\mathcal{L}} D_{ik}(f_{ik}(\lambda)) \quad (39)$$

Since each link delay D_{ik} is a convex function of the link flow, $D(\lambda)$ is convex in λ, and hence

$$\frac{dD(\lambda)}{d\lambda}\Big|_{\lambda=0} \leq D(\boldsymbol{\phi}^*) - D(\boldsymbol{\phi})$$

Since $\boldsymbol{\phi}^*$ is arbitrary, proving that $dD(\lambda)/d\lambda \geq 0$ at $\lambda = 0$ will complete the proof. From (39) to (38),

$$\frac{dD(\lambda)}{d\lambda}\Big|_{\lambda=0} = \sum_{(i,k)\in\mathcal{L}} \frac{dD_{ik}(f_{ik})}{df_{ik}} (f_{ik}^* - f_{ik}) \quad (40)$$

We now show that

$$\sum_{(i,k)\in\mathcal{L}} \frac{dD_{ik}(f_{ik})}{df_{ik}} f_{ik}^{m*} \geq \sum_{(j,k)\in\mathcal{L}} r_k^m(j) \frac{\partial D(\boldsymbol{\phi})}{\partial r_k^m(j)} \quad (41)$$

Note from (10) that

$$\sum_{k\in\mathcal{N}} D'_{ik}(t_i^m(j)\phi_{ik}^{m*}(j))\phi_{ik}^{m*}(j) \geq \frac{\partial D(\boldsymbol{\phi})}{\partial r_i^m(j)} - \sum_{k\in\mathcal{N}} \frac{\partial D(\boldsymbol{\phi})}{\partial r_k^m(j)} \phi_{ik}^*(j) \quad (42)$$

Multiplying both sides of (42) by $t^*(j)$, summing over j, and recalling (6) and (3), we have

$$\sum_{k\in\mathcal{N}} \frac{dD_{ik}(f_{ik})}{df_{ik}} f_{ik}^{m*} \geq \sum_{j\in\mathcal{N}} t_i^{m*}(j) \frac{\partial D(\boldsymbol{\phi})}{\partial r_i^m(j)} - \sum_{j,k\in\mathcal{N}} t_i^{m*}(j)\phi_{ik}^{m*}(j) \frac{\partial D(\boldsymbol{\phi})}{\partial r_k^m(j)}$$

Further summing both sides of above equation over i, we have

$$\sum_{(i,k)\in\mathcal{L}} \frac{dD_{ik}(f_{ik})}{df_{ik}} f_{ik}^{m*} \geq \sum_{i,j\in\mathcal{N}} t_i^{m*}(j) \frac{\partial D(\boldsymbol{\phi})}{\partial r_i^m(j)} - \sum_{i,j,k\in\mathcal{N}} t_i^{m*}(j)\phi_{ik}^{m*}(j) \frac{\partial D(\boldsymbol{\phi})}{\partial r_k^m(j)} \quad (43)$$

Substituting (2) into (43), we get (41). Note that if we replace $\boldsymbol{\phi}^*$ with $\boldsymbol{\phi}$ in (42), (42) becomes an equality from the equation for $\partial D/\partial r_i^m(j)$ in (7). For the same reason, if we replace $\boldsymbol{\phi}^*$ with $\boldsymbol{\phi}$ in (43), (43) becomes

$$\sum_{(i,k)\in\mathcal{L}} \frac{dD_{ik}(f_{ik})}{df_{ik}} f_{ik}^m = \sum_{j,k} r_k^m(j) \frac{\partial D(\boldsymbol{\phi})}{\partial r_k^m(j)} \quad (44)$$

Substituting (44) and (41) into (40), and summing over m, we see that $dD(\lambda)/d\lambda \geq 0$ at $\lambda = 0$, which completes the proof. ■

APPENDIX D: PROOF OF LEMMA 2

Proof: (a) Assume that $\boldsymbol{\phi}^{(k+1)}$ is not loop-free so that there exists a sequence of links forming a directed cycle along which $\boldsymbol{\phi}^{(k+1)}$ is positive. Also there must exist a link (p,q) for which $\frac{\partial D(\boldsymbol{\phi}^{(k)})}{\partial r_p^m(j)} \leq \frac{\partial D(\boldsymbol{\phi}^{(k)})}{\partial r_q^m(j)}$. From the definition of $B_{i,\boldsymbol{\phi}^{(k)}}^m(j)$ we must have $\phi_{pq}^m(j)^{(k)} > 0$ and hence (p,q) is an improper link. Now move backwards around the cycle to the first link (i,l) for which $\phi_{il}^m(j)^{(k)} = 0$. Such a link must exist since $\boldsymbol{\phi}^{(k)}$ is loop-free. Since node l is upstream of node p and link (p,q) is improper, we have $l \in B_{i,\boldsymbol{\phi}^{(k)}}^m(j))$ which contradicts the hypothesis $\phi_{il}^m(j)^{(k+1)} > 0$.

(b) If $\Delta\boldsymbol{\phi}^{(k)} = 0$ solves problem (26), then we must have $\boldsymbol{\delta}_i^m(j)^T \Delta\boldsymbol{\phi}_i^m(j) \geq 0$ for each node i and $\Delta\boldsymbol{\phi}^m(j)$ satisfying the constraints of (26).

$$\Delta\boldsymbol{\phi}_i^m(j) \geq -\boldsymbol{\phi}_i^m(j)^{(k)}, \sum_{l\in\mathcal{N}} \Delta\phi_{il}^m(j) = 0, \forall l \in B_{i,\boldsymbol{\phi}^{(k)}}^m(j)$$

By writing $\Delta\boldsymbol{\phi}_i^m(j) = \boldsymbol{\phi}_i^m(j) - \boldsymbol{\phi}_i^m(j)^{(k)}$ and using (7), (16) we have

$$\begin{aligned} & \boldsymbol{\delta}_i^m(j)^T(\boldsymbol{\phi}_i^m(j) - \boldsymbol{\phi}_i^m(j)^{(k)}) \\ = & \sum_{l\in\mathcal{N}} \delta_{il}^m(j)\phi_{il}^m(j) - \sum_{l\in\mathcal{N}} \delta_{il}^m(j)\phi_{il}^m(j)^{(k)} \\ = & \sum_{l\in\mathcal{N}} \delta_{il}^m(j)\phi_{il}^m(j) - \frac{\partial D}{\partial r_i^m(j)} \geq 0 \end{aligned}$$

By considering $\phi_{il}^m(j) = 1$ for each $l \notin B_{i,\boldsymbol{\phi}^{(k)}}^m(j)$, we obtain

$$\frac{\partial D}{\partial r_i^m(j)} \leq \delta_{il}^m(j), \forall l \notin B_{i,\boldsymbol{\phi}^{(k)}}^m(j)$$

From (7) and (16) we have

$$\frac{\partial D}{\partial r_i^m(j)} = \delta_{il}^m(j), \forall l \notin B_{i,\boldsymbol{\phi}^{(k)}}^m(j), \phi_{il}^m(j)^{(k)} > 0$$

Since $D'_{il} > 0$ for all $(i,l) \in \mathcal{L}$ it follows from (7), (16) and the relation above that there are not improper links, and using the definition of $B_{i,\boldsymbol{\phi}^{(k)}}^m(j)$ we obtain

$$\frac{\partial D}{\partial r_i^m(j)} = \min_{l \in \mathcal{N}} \delta_{il}^m(j)$$

which is the same as (10), the sufficient condition for optimality of $\boldsymbol{\phi}^{(k)}$.

(c) If $\boldsymbol{\phi}^{(k)}$ is optimal then from the necessary condition for optimality (9) we have that for all node i with $t_i^m(j) > 0$

$$\frac{\partial D}{\partial r_i^m(j)} = \min_{p \in \mathcal{N}} \delta_{ip}^m(j)$$

It follows using a reverse argument to the one in (b) that $\Delta\boldsymbol{\phi}_i^m(j)^{(k)} = 0$ if $t_i^m(j) > 0$. Since changing only routing variables of nodes i for which $t_i^m(j) = 0$ does not affect the flow through each link we have $D(\boldsymbol{\phi}^{(k)}) = D(\boldsymbol{\phi}^{(k+1)})$ and $\boldsymbol{\phi}^{(k+1)}$ is optimal.

(d) If $t_i^m(j) > 0$, then $\boldsymbol{M}_i^m(j)^{(k)}$ is positive definite on the appropriate subspace. If in addition $\Delta\boldsymbol{\phi}_i^m(j)^{(k)} \neq 0$, then the second term in (26) is positive. Since the minimum in (26) is non-positive, $\boldsymbol{\delta}_i^m(j)^T \Delta\phi_i^m(j)^{(k)} < 0$. By (16), we obtain that

$$\Big(\frac{\partial D}{\partial \boldsymbol{\phi}_i^m(j)}\Big)^T \Delta\boldsymbol{\phi}_i^m(j)^{(k)} < 0$$

Hence $\Delta\boldsymbol{\phi}^{(k)}$ is a direction of descent at $\boldsymbol{\phi}^{(k)}$ and the result follows. ∎

Appendix E: Proof of Theorem 3

Proof: We only give the sketch of the proof due to space constraint. Since the delay function D_{ik} is bounded from below by 0, showing that the sequence $\{D(\boldsymbol{\phi}^{(k)})\}_{k=1}^{\infty}$ is nonincreasing implies that

$$\lim_{k\to\infty} (t_i^m(j)^{(k)})^2 \cdot \|\boldsymbol{\phi}_i^m(j)^{(k+1)} - \boldsymbol{\phi}_i^m(j)^{(k)}\|^2 = 0 \quad (45)$$

which implies convergence of the sequence to $\min_{\boldsymbol{\phi}\in\psi} D(\boldsymbol{\phi})$.

In order to prove (45), we first prove that

$$\begin{aligned} &\sum_{i,j\in\mathcal{N}} t_i^m(j)^{(k)} t_i^m(j)^{(k+1)} \|\boldsymbol{\phi}_i^m(j)^{(k+1)} - \boldsymbol{\phi}_i^m(j)^{(k)}\|^2 \\ \geq\ & p \sum_{i,j\in\mathcal{N}} (t_i^m(j)^{(k)})^2 \|\boldsymbol{\phi}_i^m(j)^{(k+1)} - \boldsymbol{\phi}_i^m(j)^{(k)}\|^2 \quad (46) \end{aligned}$$

where p is a positive scalar.

Using (46), we are able to prove that

$$\begin{aligned} &D(\boldsymbol{\phi}^{(k+1)}) - D(\boldsymbol{\phi}^{(k)}) \leq (\frac{\lambda}{\alpha|\mathcal{N}|^3} + \overline{U}|\mathcal{N}|^4) \cdot \\ &\sum_{i,j\in\mathcal{N}} (t_i^m(j)^{(k)})^2 \|\boldsymbol{\phi}_i^m(j)^{(k+1)} - \boldsymbol{\phi}_i^m(j)^{(k)}\|^2 \\ &\forall i,j \in \mathcal{N}, m \in \mathcal{M} \end{aligned}$$

where $\overline{U} = \max_{(i,l)\in\mathcal{L},\boldsymbol{\phi}\in\psi}\{D''_{ik}(f_{il}^{(k)}) \mid D(\boldsymbol{\phi}^{(k)}) \leq D_0\}$. Take $\alpha \in (0, \overline{\alpha}]$, $\overline{\alpha} < \frac{\lambda}{|\mathcal{N}|^7\overline{U}}$ we have

$$\begin{aligned} &D(\boldsymbol{\phi}^{(k+1)}) - D(\boldsymbol{\phi}^{(k)}) \leq -p \sum_{i,j} (t_i^m(j)^{(k)})^2 \cdot \\ &\|\boldsymbol{\phi}_i^m(j)^{(k+1)} - \boldsymbol{\phi}_i^m(j)^{(k)}\|^2, \forall i,j \in \mathcal{N}, m \in \mathcal{M} \end{aligned}$$

Thus, the sequence $\{D(\boldsymbol{\phi}^{(k)})\}_{k=1}^{\infty}$ is nonincreasing. Since D_{ik} is bounded from below by 0, we obtain (45). ∎

References

[1] R. Ahlswede, N. Cai, S.R. Li, and R.W. Yeung, "Network information flow," *IEEE Tran. Information Theory*, vol. 46, 2000.

[2] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE Tran. Networking*, vol. 11, 2003.

[3] R. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Tran. Commun.*, vol. 25, 1977.

[4] D. Bertsekas, E. Gafni, and R. Gallager, "Second derivative algorithms for minimum delay distributed routing in networks," *IEEE Tran. Commun.*, vol. 32, 1984.

[5] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1979.

[6] M. Thimm, "On the approximability of the steiner tree problem," in *Mathematical Foundations of Computer Science*. 2001, Springer LNCS 2136.

[7] G. Robins and A. Zelikovsky, "Improved steiner tree approximation in graphs," in *Proc. of 7th ACM-SIAM Symp. on Discrete Algorithms*, 2000.

[8] K. Jain, M. Mahdian, and M. Salavatipour, "Packing steiner trees," in *Proc. of 10th ACM-SIAM Symp. on Discrete Algorithms*, 2003.

[9] Y. Chu, R. Rao, and H. Zhang, "A case for end system multicast," in *ACM SIGMETRICS*, 2000.

[10] S.R. Li, R.W. Yeung, and N. Cai, "Linear network coding," *IEEE Tran. Information Theory*, vol. 49, 2003.

[11] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *Allerton Conference on Communication, Control, and Computing*, 2003.

[12] L, Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill, 1964.

[13] C. Cassandras, M. Abidi, and D. Towsley, "Distributed routing with on-line marginal delay estimation," *IEEE Tran. Commun.*, vol. 38, 1990.

[14] M. Kodialam and T. Nandagopal, "Characterizing the achievable rates in multihop wireless networks," in *ACM MOBICOM*, 2003.

[15] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu , "Impact of interference on multi-hop wireless network performance," in *ACM MOBICOM*, 2003.